Agenda

- Benefits
- How we view standardization
- The issues
- Where are we?

Benefits of Standardization

- From the 10/5/01 CSAC presentation
  - Reduce development and support costs
  - Economies of scale
  - Better support for collaboration (data exchange)
  - Productivity when moving between organizations
- With these benefits in mind, how do we view standardization

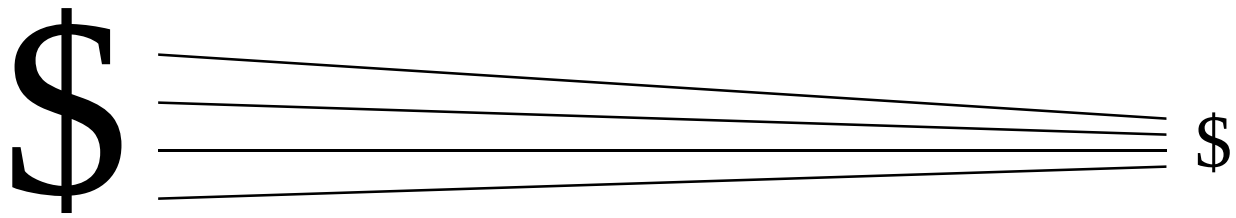
How We View Standards

- Issues are different for UNIX systems though there are some analogies in the Novell/NT file server area.
- Not simply choosing a standard computer.
- Many other issues affecting interoperability, sharing of data, productivity, and cost.

The Issues

Complete Freedom
Of Choice

Strict
Standardization



No standards

Basic
Interoperability

Uniform
Environment

Configuration
Management

Standardized
OS load and
Hardware

The Basics – Interoperability Issues

UID and GID Namespace

- UID – a user ID number and login name
- GID – a group ID used for group level permissions
- Need to define UID and GID namespace
- Avoid UID and GID clashes
- Required for collaborations requiring NFS sharing of data.
- Required for group programming environments.
- File level security implications
- Can have long term impact. (e.g. tape archived files are referenced by UID)
- Currently handled by web UID and GID registration

The Basics – cont.

- Time synchronization
  - —Important for collaborations doing software development
  - —This issue is often overlooked by the users.
  - —Recommend a standard Network Time Protocol configuration.
- AFS vs. NFS
  - —Global file system namespace
  - —Uniform file system naming
  - —Affects sharing of file systems.
  - —Making the correct choices can facilitate (or hinder) collaboration with other groups.

Uniform Environment

- Uniform programming environment for user
  - Is your programming environment (e.g. existence and location of tools, system programs, shared libraries) the same across systems?
  - Will your programs work if simply copied to another system?
- How are new software and tools installed across systems?
  - Compilation problems, shared library conflicts.
  - Is there a choice of software?
  - Ability to try new software w/o affecting entire system?

Uniform Environment – cont.

How do we currently address these issues?

- Software Farm
  - Central server that provides common software and tools.
  - Provides consistent path across systems and different platforms.
  - Accessed in a way that does not require reconfiguration.
- Modules
  - Provides interface to dynamically load/unload software in a users environment.
  - Granular choice of software – users can pick and choose w/o affecting others on the system.
  - Abstracts software installation details from user.
  - Also used on the NERSC supercomputers.
- Currently about 50% of the UNIX systems on LBLNet utilize the Software Farm and Modules

Configuration Management

- Determines how effectively we can manage a large number of systems.
- Extremely important for doing updates and managing security.
- Methods:
  - —Ad hoc
  - —Centralized Push – Database driven
    - GNU cfengine
    - Tivoli
  - —Centralized Pull/Analyze/Push
    - CSGROUP SLURP Engine

Standard OS Load

- Currently there is not a standard UNIX OS load.
- Feasible if there is a demand for carbon copy systems.
- NERSC Division has accomplished this with some success.

Standard OS Load – cont.

The general trend is towards Linux. If we were to do a standard load, it would be for Linux.

	Jan 2000	Jan 2002
Sun	803	550
Linux	100 est.	300
SGI	119	100

Notes:

- Older Suns have been retired.
- Linux clusters are counted as one network entity so cluster nodes are not included in this number.
- SGIs are still in use to support specialized applications.

Standard OS Load – cont.

What are some issues with doing a standard Linux load?

- What packages should be included in the "standard build"?
  - Clearly, most distributions come with the kitchen sink - what do users need?
  - Should there be different groups of packages for different needs?
- How often should the "build" be updated?
  - RPMs come out every day.
  - How should the builds be "revisioned"?

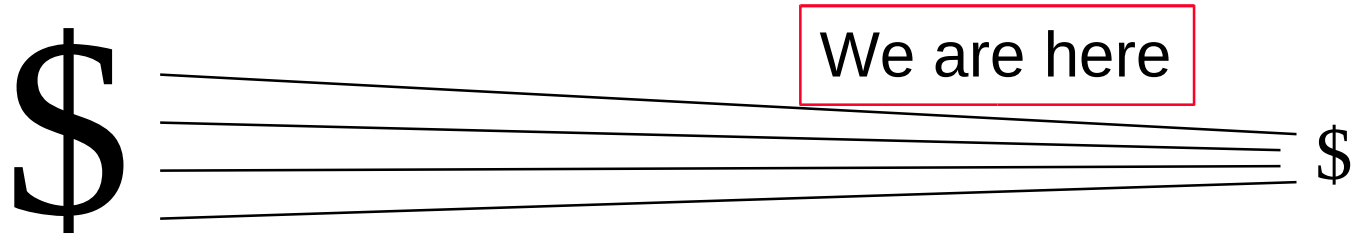
Standard Hardware

- One size fits all, but must be meet the needs of the scientific user.
- Can facilitate the implementation of standard OS loads to some degree.
- Hardware choice does not necessarily affect productivity or cost of support. The following don't really matter on a UNIX system. (as long as sufficient and compatible)
  - CPU vendor (AMD vs. Intel) and speed (Mhz)
  - Disk size
  - Memory
  - Motherboard/Processor chipset (Intel, SIS, VIA, AMD)

Where are we?

Complete Freedom Of Choice

Strict Standardization



No standards

Basic Interoperability

Uniform Environment

Configuration Management

Standardized OS load and Hardware